

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKA TEADUSKOND
Arvutiteaduse instituut
Informaatika õppekava

Katrin Valdson

Tekstülesannete lahendamise programmi täiendamine

Bakalaureusetöö (6 EAP)

Juhendaja: dotsent Rein Prank

Tartu 2014

Tekstülesannete lahendamise programmi täiendamine

Lühikokkuvõte:

Tekster on 2006. aastal loodud ning 2013. aastal täiendatud algklassidele mõeldud tekstülesannete lahendamise rakendus, mis koosneb ülesannete koostamise ja lahendamise programmidest. Käesolevas töös antakse ülevaade Tekster-iga sarnastest programmidest, selle 2013. aasta versioonist ning antud bakalaureuse töö käigus tehtud muudatustest, millest kõige olulisem oli ülesannete automaatlahenduse võimaldamine.

Võtmesõnad:

Tekstülesanne, õpiprogramm, automaatlahendus.

Adding attributes to an existing word problem solving program

Abstract:

Tekster is a word problem solving program meant for primary school students which was created in 2006 and continued in 2013. The program consists of the word problem solving and creating programs. In this paper, an overview is given of programs which are similar to Tekster, the 2013. year version and the changes made during the current thesis, out of which the most important part was enabling the automated solving of the word problems.

Keywords:

Word problem, educational software, automated problem solving.

Sisukord

Sissejuhatus.....	4
1. Sarnaseid programme.....	5
1.1. Carnegie Learning - MATHia.....	5
1.1.1. MATHia võrdlus Tekster-iga.....	7
1.2. Mobile Games - Thinking Blocks.....	8
1.1.2. Thinking Blocks-i võrdlus Tekster-iga.....	10
2. Programmi varasema versiooni kirjeldus.....	11
2.1. Ülesande lahendamine.....	12
2.1.1. Küsimuse moodustamine.....	12
2.1.2. Avaldise moodustamine.....	13
2.1.3. Tehte väärtuse arvutamine.....	15
2.1.4. Lahendusfail.....	17
2.2. Ülesande koostamine.....	18
2.2.1. Ülesande teksti sisestamine.....	18
2.2.2. Küsimuste moodustamine.....	19
2.2.3. Küsimustele vastavate avaldiste määramine.....	20
2.2.4. Ülesande esitus tekstifailis.....	22
3. Bakalaureusetöö käigus tehtud muudatused.....	24
3.1. Maxima.....	24
3.2. Ülesande automaatne lahendamine.....	25
3.2.1. Automaatlahenduse rakendamine ülesannete lahendamise programmis.....	27
3.2.2. Automaatlahenduse rakendamine ülesannete koostamise programmis.....	28
3.3. Sammude kustutamine.....	30
4. Programmi edasiarendamise võimalused.....	31
Kokkuvõte.....	32
Tsiteeritud teosed.....	33
Lisad.....	34
I. Projekt.....	34
II. Litsents.....	35

Sissejuhatus

Käesoleva bakalaureusetöö sisuks on tekstülesannete lahendamise programmi Tekster võimaluste laiendamine. Tekster-i baasversiooni on loonud Evari Koppel 2006. aastal [1] ning selle täiendatud versiooni Joosep Kibal 2013. aastal bakalaureusetöö [2] raames.

Käesolevas töös kirjeldatakse Tekster-i 2013. aasta versiooni lisatud täiendusi, millest peamiseks on automaatlahenduse programmeerimine.

Töö koosneb neljast peatükist. Esimeses peatükis tuuakse välja Tekster-iga sarnaseid programme ning nende erinevusi võrreldes Tekster-iga. Teises peatükis kirjeldatakse programmi 2013. aasta versiooni. Kolmandas peatükis tutvustatakse programmile tehtud muudatusi. Neljandas peatükis tuuakse välja võimalused, kuidas rakendust edasi arendada.

Lisana on esitatud täiendatud programmid ja ülesannete kogu.

1. Sarnaseid programme

Järgnevalt on kirjeldatud teisi tekstülesannete lahendamise programme ning võrreldud neid Tekster-iga.

1.1. Carnegie Learning - MATHia

Carnegie Learning - MATHia [3] on ingliskeelne tekstülesannete lahendamise veebirakendus, mis on mõeldud 6.-8. klasside õpilastele. Pärast programmi kodulehel vastava ankeedi täitmist isikuandmetega, saadetakse sisestatud e-maili aadressile kasutajanimi ning parool, millega on võimalik kasutada 90-päevast tasuta prooviversiooni. Prooviversioonis on ligipääsu programmi mõnele võimalusele piiratud. MATHia programmil on nii õpetaja kui ka õpilase rakendus.

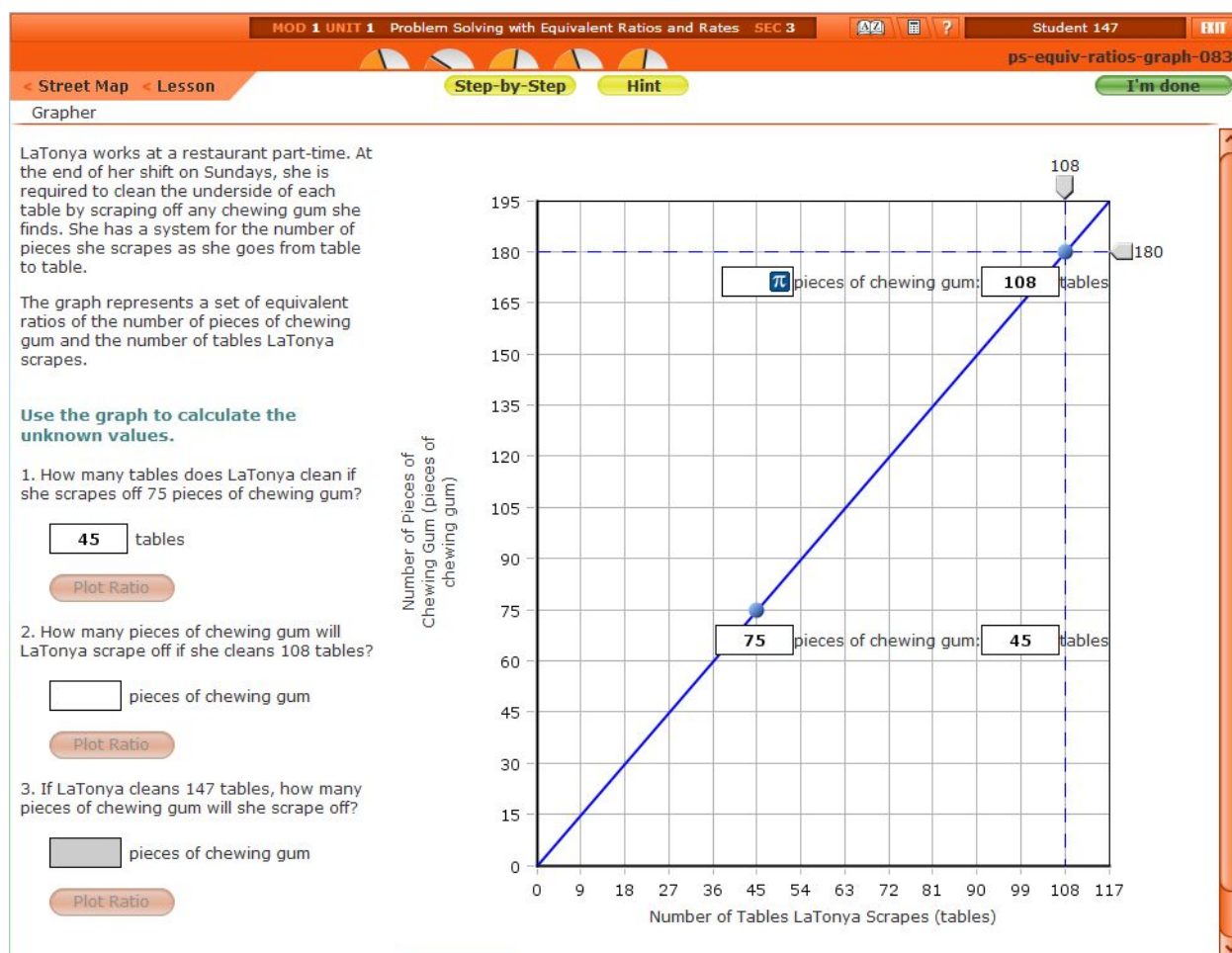
Õpilase rakenduses toimub ülesannete lahendamine. Ülesanded on jagatud moodulitesse, mis koosnevad peatükkidest. Mooduli iga peatüki ülesannete lahendamine toimub järgmiselt:

- 1) Õpilasel on võimalik lugeda käesoleva peatüki ülesannete lahendamise juhendit;
- 2) Lahendajal on võimalik läbida näiteülesanne, milles programm näitab samm-sammult, kuidas käesoleva peatüki ülesandeid lahendada ning kuidas korrektselt vastuseid sisestada;
- 3) Õpilane liigub edasi iseseisvalt ülesannete lahendamise juurde. Lahendamise käigus jälgib programm, milliseid vigu kasutaja teeb, ning valib järgmised ülesanded vastavalt sellele. Lahendamise käigus on võimalik programmilt abi küsida;
- 4) Kui lahendaja on korrektselt käesoleva peatüki ülesanded lahendanud, loetakse peatükk läbituks ning võimaldatakse alustada järgmisega.

Õigete lahenduste eest saavad kasutajad auhindu ning võivad muuta rakenduse välimust, näiteks värvide muutmise ja kujunduse teema valimine.

Järgnevalt (Joonis 1) on näha ülesande iseseisva lahendamise vaade. Akna vasakus osas on ülesande tekst, küsimused, millele lahendaja peab vastama, ning nende all lahtrid, kuhu sisestatakse küsimuse vastus. Akna paremas osas on graafik, mille kumbki telg esindab ülesande jaoks olulist muutujat, ning mille abil on võimalik ülesannet lahendada. Graafikul asuvaid ringikujulisi nuppe saab lohistada soovitud punkti ning seejärel leida antud punkti koordinaadid. Nupule "Hint" vajutades annab programm kasutajale teada, mis tegevust on soovitatav järgmisena

teha. Kui kasutaja on ülesande lõpuni lahendanud, on võimalik vajutada nupule “I’m done”, mis annab lahendajale uue ülesande.

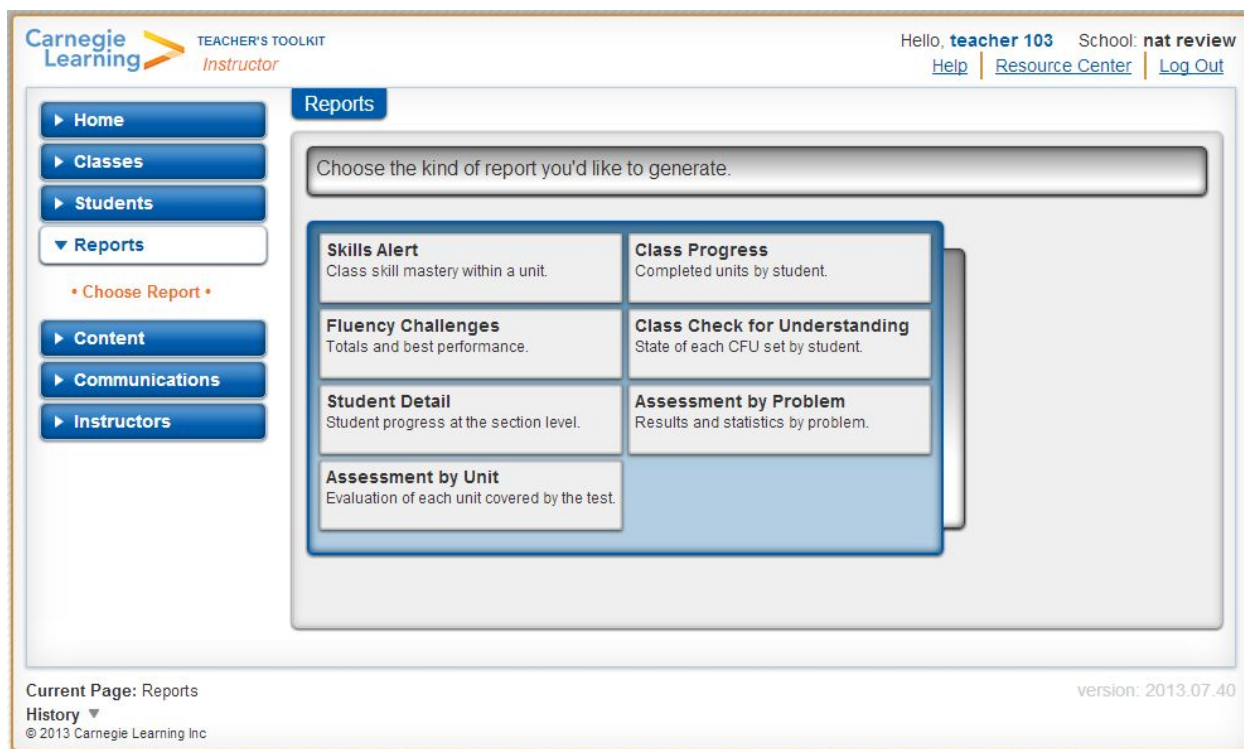


Joonis 1. Ülesande lahendamise vaade.

Õpetaja rakenduses on võimalik teha järgnevaid tegevusi:

- 1) Kuvada vastavale õpetajale määratud ainete nimekirja ning neid juurde lisada;
- 2) Vaadata õpilaste nimekirja ja õpilasi juurde lisada;
- 3) Kuvada statistikat õpilaste ülesannete lahendamise edukusest (Joonis 2), näiteks terve klassi tulemused mingis õppetükis, parimad tulemused, kindla ülesandetüübi tulemused klassis;
- 4) Näha olemasolevaid mooduleid ning materjali ning neid juurde lisada;
- 5) Vaadata saabunud kirju ning kalendrit;
- 6) Vaadata õpetajate ja juhendajate nimekirja ning tegevusi.

Järgneval joonisel (Joonis 2) on näha õpetaja vaade, kui õpetajale võimaldatud tegevustest on valitud ülesannete lahendamise statistika vaatamine. Ekraani paremas ääres on näha ülejäänud tegevused, mida õpetaja saab sooritada. Ekraani keskel on näha erinevad võimalused, kuidas õpetaja õpilaste ülesannete lahendamise statistikat saab kuvada.



Joonis 2. Õpetajaprogrammi statistika vaade.

1.1.1. MATHia võrldus Tekster-iga

MATHia võimalused, mis Tekster-is puuduvad:

- 1) Ülesannetele illustreeriva ning abistava materjali lisamine, näiteks graafikud ning tabelid;
- 2) Õpilasele ülesannete püstitamine, millega sarnaste lahendamisel on tekkinud vigu.
- 3) Ülesande lahendamiseks vajalike tegevuste ettenäitamine, näiteks millisele nupule vajutada järgmisena. Tekster-i puhul ütleb juhend, mida peaks järgmisena tegema, kuid ei näita, kuidas seda teha;
- 4) Lahendaja premeerimine õigete lahenduste eest;
- 5) MATHia-t ei ole vaja kasutamiseks arvutisse paigaldada, kuna tegemist on veebirakendusega;
- 6) Tekster-i õpetajaprogrammist puuduvad kõik MATHia õpetajaprogrammi võimalused, väljaarvatud ülesannete koostamine.

Tekster-i võimalused, mis MATHia programmil puuduvad:

- 1) Kasutaja poolt küsimuste moodustamine. MATHia-s on küsimused koostatud ning lahendaja ülesanne on neile vaid vastata;
- 2) Näha detailselt, kuidas õpilane on ülesande lahendanud. MATHia-s on võimalik kuvada vaid statistikat ning kuna ülesannete lahendamine põhineb peamiselt vastuste sisestamisel illustreeriva materjali abiga, siis ei ole lahendaja mõttekäik ülesande lahendamisel õpetajale nähtav;
- 3) Võimalus lahendajal ise otsustada, millist ülesannet järgmisena lahendada;
- 4) Tekster-i kasutamine on võimalik ka internetiühenduseta.

1.2. Mobile Games - Thinking Blocks

Thinking Blocks [4] on ingliskeelne tekstülesannete lahendamise vabavaraline veebirakendus, mis on mõeldud 1.-6. klasside õpilastele. Programm koosneb neljast moodulist: liitmine ja lahutamine, korrutamine ja jagamine, murrud ning suhted ja proportsioonid. Iga moodul on jagatud kuueks ülesandegrupiks, millest kasutaja võib lahendada ülesandeid omal valikul.

Iga ülesande lahendamine toimub järgmiselt:

- 1) Valitakse moodul;
- 2) Moodulist valitakse sobiv ülesannete grupp;
- 3) Soovi korral on võimalik vaadata vastava grupi ülesannete lahendamise õpetust video kujul;
- 4) Otsustatakse, kas hakatakse arvutama arvudega 1-50 või 1-300;
- 5) Lahendaja koostab mudeli ülesande andmetest (Joonis 3), mis aitab ülesannet visualiseerida;
- 6) Õpilane kannab koostatud mudelile ülesande tekstis etteantud väärtused ning tuvastab väärtuse, mis on vaja leida (Joonis 4);
- 7) Lahendaja arvutab ülesande vastuse kas kalkulaatoriga või ilma. Programm annab jooksvalt õpilasele teada, mis tegevus on vaja järgmisena teha ning ka täpsemaid vihjeid, kui on valesti vastatud.

Igas ülesandetüübi grupis tuleb korrektselt lahendada viis ülesannet, et grupp loetaks läbituks. Thinking Blocks rakendusel puudub õpetaja vaade.

Järgnevalt (Joonis 3) on näha ülesande lahendamise etapp, mil on vaja moodustada ülesandes antud andmete kohta mudel. Ekraani ülemises ääres on näha ülesande tekst, alumises ääres teavitatakse kasutajat tegevusest, mida peaks hetkel sooritama. Alumises paremas nurgas näidatakse kasutajale, mitu ülesannet ta edukalt on lahendanud. Lillat värvi kastid tähendavad, et lahendatud pole veel ühtegi. Ekraani paremas servas on sildid, mis tuleb mudelile kanda, ning mis esindavad ülesandes antud andmeid ning väärtust, mis on vaja leida. Vaate keskel on mudel, millele tuleb sildid korrektselt kanda.

Word Problem: Tommy sells drinks at the ball park. He sold 6 drinks during the first inning and 11 drinks during the second inning. If Tommy had 9 drinks left to sell, how many did he have at first?

Build a model showing the parts and the whole.

drinks at first

first inning

Label

Label

Check

amount left

second inning

Instructions
Read the word problem. Identify the parts and whole. Then build a model.
Drag 3 blocks and 4 labels to the targets in the work area.
Tap the **Check** button to check your work.

Part Whole Model - 3 Parts

Joonis 3. Mudeli koostamine.

Kui mudel on korrektselt koostatud, avaneb kasutajale järgnev vaade (Joonis 4). Antud vaates on ekraani paremas servas väärtused, mis tuleb vastavalt ekraani keskel olevale eelnevalt koostatud mudelile lohistada.

Word Problem: Tommy sells drinks at the ball park. He sold 6 drinks during the first inning and 11 drinks during the second inning. If Tommy had 9 drinks left to sell, how many did he have at first?

Add numbers to your model. Use a ? to show the missing number.

drinks at first

?

6

first inning

second inning

amount left

Check

9

11

Feedback
Excellent work! Blocks and labels have been placed correctly. Now you can add numbers to the model.

Part Whole Model - 3 Parts

Joonis 4. Mudelile väärtuste kandmine.

1.1.2. Thinking Blocks-i võrdlus Tekster-iga

Thinking Blocks-i võimalused, mis Tekster-is puuduvad:

- 1) Abistavate mudelite koostamine;
- 2) Rakendusesiseselt kalkulaatori kasutamine;
- 3) Kasutaja tegevustele, näiteks nupu vajutamine, tagasiside andmine heliefektidega;
- 4) Harilike murdudega arvutamine;
- 5) Mudelile märkmete tegemine joonistamise tööriistaga;
- 6) Pole vajadust programmi kasutaja arvutisse paigaldada, kuna tegemist on veebirakendusega;
- 7) Lahendajale vastuse näitamine, kui valesti on vastatud kolm korda.

Tekster-i võimalused, mis Thinking Blocks-is puuduvad:

- 1) Kasutaja poolt küsimuste moodustamine;
- 2) Vahevastuste arvutamine. Thinking Blocks nõuab vaid lõppvastust;
- 3) Ülesande lahenduskäigu salvestamine;
- 4) Õpetajaprogramm, ülesannete koostamine;
- 5) Tekster ei nõua interentiühenduse loomist.

2. Programmi varasema versiooni kirjeldus

Tekster-i ülesannete lahendamise programm on mõeldud algklassidele tekstülesannete samm-sammult lahendamiseks. Näiteks olgu ülesanne järgmine:

Teises klassis õpib 8 poissi ja 9 tüdrukut.

Kolmandas klassis õpib 7 poissi ja 10 tüdrukut.

Iga õpilane saab lõunaks 2 mandariini.

Mitu mandariini said õpilased kokku?

Õpilane võib programmi abil lahendada ülesande järgneval viisil:

1. Mitu õpilast õpib teises klassis? $8+9=17$
2. Mitu õpilast õpib kolmandas klassis? $7+10=17$
3. Mitu õpilast õpib teises ja kolmandas klassis? $17+17=34$
4. Mitu mandariini saavad õpilased kokku? $34*2=68$

Võimalik on lahendada ülesannet ka teistmoodi:

1. Mitu tüdrukut õpib teises ja kolmandas klassis? $9+10=19$
2. Mitu poissi õpib teises ja kolmandas klassis? $8+7=15$
3. Mitu õpilast õpib teises ja kolmandas klassis? $19+15=34$
4. Mitu mandariini saavad õpilased kokku? $34*2=68$

Õpilane võib aga ülesannet lahendada ka viisil, mis ei ole optimaalne:

1. Mitu õpilast õpib teises klassis? $8+9=17$
2. Mitu õpilast õpib kolmandas klassis? $7+10=17$
3. Mitu mandariini said õpilased teises klassis? $17*2=34$
4. Mitu mandariini said õpilased kolmandas klassis? $17*2=34$
5. Mitu mandariini saavad õpilased kokku? $34+34=68$

Järgnevalt kirjeldatakse, kuidas toimub tekstülesannete lahendamine ja koostamine Tekster-i 2013. aasta versioonis.

2.1. Ülesande lahendamine

Tekster-i ülesannete lahendamise programmi avamisel küsitakse lahendaja nime. Sisestatud nime järgi antakse nimi lahendusfailile kujul “<ülesande nimi>_<lahendaja nimi>.txt”, näiteks “Ülesanne1_Katrin.txt”.

Tekster-i ülesannete põhiideeks on lahenduse leidmine, mis koosneb sammudest, kus igal sammul leitakse ühe tehte abil järjekordne ülesande jaoks tähtsust omav suurus. Ülesande lahendamise iga samm koosneb kolmest etapist: küsimuse moodustamine, küsimusele vastava tehte moodustamine juba teadaolevatest arvudest ning tehte tulemuse arvutamisest ja kontrollist.

2.1.1. Küsimuse moodustamine

Kasutaja eesmärgiks on moodustada küsimus, millele saab vastata ühe tehtega. Lauseosad on jagatud veergudesse, mis kokku moodustavad küsimustegrupi, ning küsimuse koostamiseks tuleb igast veerust valida üks lauseosa. Võimalik on koostada ka ebasobivaid küsimusi.

Küsimusi saab moodustada maksimaalselt kahes grupis. Küsimustegrupi ees oleva teksti “Vali” ette on võimalik panna linnuke, mis muudab vastava grupi aktiivseks ning teise grupi mitteaktiivseks. Ainult valitud grupis on võimalik küsimust moodustada. Kui küsimus on koostatud, vajutab kasutaja nupule “Kontrolli küsimust” (Joonis 5).

Tekster - tekstülesannete lahendamise programmi baasvariant

File Programmist

Ülesanne- mandariinid

Teises klassis õpib 8 poissi ja 9 tüdrukut. Kolmandas klassis õpib 7 poissi ja 10 tüdrukut. Iga õpilane saab lõunaks 2 mandariini. Mitu mandariini said õpilased kokku?

☒ Vali ☐ poissi ☒ teises klassis? ☐ kolmandas klassis? ☐ teises ja kolmandas klassis?

☒ Mitu ☐ tüdrukut ☒ õpib ☐ õpilast

☐ Vali ☐ Mitu mandariini said ☐ tüdrukud ☐ kolmandas klassis? ☐ õpilased ☐ teises ja kolmandas klassis?

1. küsimus Mitu õpilast õpib teises klassis?

Juhised

Koosta küsimus, mille vastuse saaks arvutada ühe tehtega.

Joonis 5. Küsimuse koostamine

Võimalikud vead, mida kasutaja võib küsimuse koostamisel teha on järgmised:

- 1) Küsimusel ei ole ülesande lahendamise vaatepunktist sisu ehk sellele ei ole ülesande tekstifailis määratud vastavat avaldist;
- 2) Küsimusele on vastus juba olemas;
- 3) Küsimusele ei saa ühe tehtega vastust leida.

Kui kontrollimisel selgub, et kasutaja on teinud mõne ülalmainitud vigadest, siis annab programm sellest märku vastava veateatega. Kui moodustatud küsimus vastab nõuetele, liigutakse järgmise etapi juurde.

2.1.2. Avaldise moodustamine

Tehte moodustamiseks tuleb ülesande tekstist või eelnevalt vastatud küsimuste vastustest lohistada väärtus selleks ettemääratud kastidesse. Kasutatud väärtuse taust muutub kollaseks. Tehe saab koosneda vaid kahest väärtusest ning ühest tehtemärgist. Tehtemärgi valimiseks tuleb vajutada

nupu peale, millel asub soovitud tehtemärk (Joonis 6).

Joonis 6. Tehte moodustamine.

Võimalikud vead, mida kasutaja võib tehte koostamisel teha on järgmised:

- 1) tehtemärgi valimata jätmine;
- 2) kummagi väärtuse valimata jätmine;
- 3) tehte moodustamine, mis ei vasta käesolevale küsimusele, mis tähendab, et ülesande tekstifailis on sellele küsimusele vastavusse määratud teistsugune avaldis.

Kui kontrollimisel selgub, et kasutaja on teinud mõne ülalmainitud vigadest, siis annab programm sellest teada vastava veateatega. Kui moodustatud tehtes vigu ei esine, liigutakse edasi tehte tulemuse arvutamise juurde.

2.1.3. Tehte väärtuse arvutamine

Tehte väärtuse arvutamiseks peab kasutaja selleks ettenähtud lahtrisse sisestama tehte tulemuse ning seejärel vajutama nuppu “Kontrolli vastust” (Joonis 7).

Tekster - tekstülesannete lahendamise programmi baasvariant

File Programmist

Ülesanne- mandariinid

Teises klassis õpib 8 poissi ja 9 tüdrukut. Kolmandas klassis õpib 7 poissi ja 10 tüdrukut. Iga õpilane saab lõunaks 2 mandariini. Mitu mandariini said õpilased kokku?

1. küsimus Mitu õpilast õpib teises klassis?

8 + 9 =

Kontrolli vastust Tagasi

Juhised

Leia vastus ning sisesta see lahtrisse.

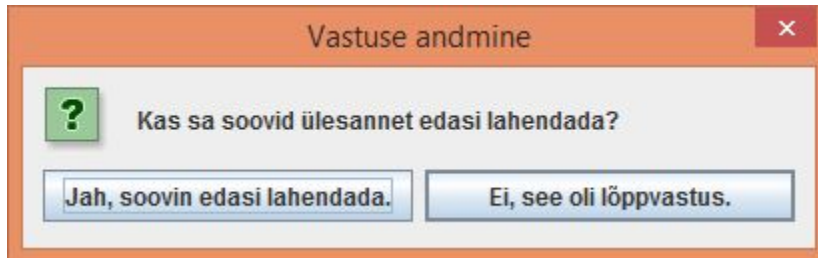
Joonis 7. Avaldise väärtuse arvutamine.

Lahtrisse on võimalik sisestada vaid numbreid ning punkti, mis tähistab komakohta. Seega vead, mida kasutaja saab teha on järgmised:

- 1) vastuse sisestamata jätmine;
- 2) valesti ümardamine;
- 3) vale vastuse sisestamine,
- 4) vastuse sisestamine, mis sisaldab rohkem kui üht punkti.

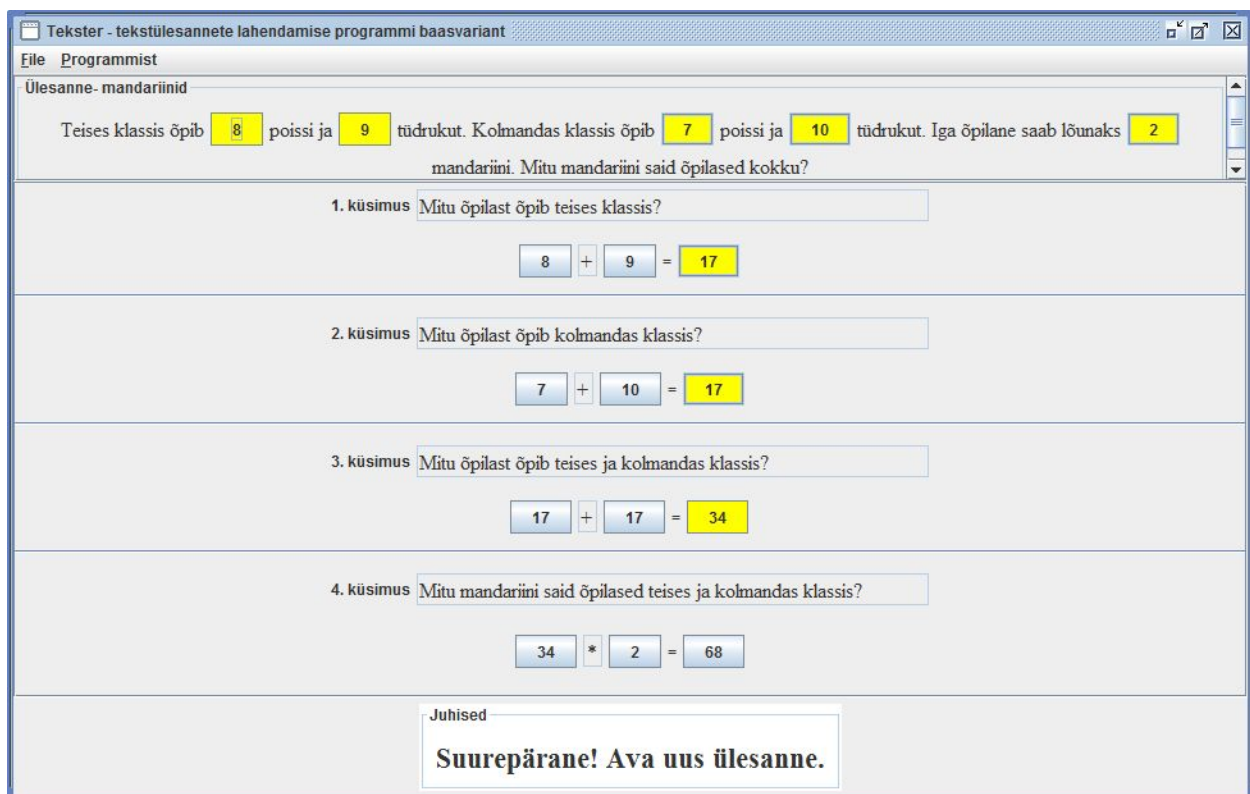
Iga vea korral annab programm sellest teada vastava veateatega.

Kui programm on sisestatud vastuse aktsepteerinud, tuleb valida, kas saadud vastus oli ülesande lõppvastus või mitte (Joonis 8). Selleks tuleb esilekerkivas dialoogiaknas vajutada vastavale nupule.



Joonis 8. Valik, kas edasi lahendada või lõpetada.

Kui lahendaja valib ekslikult, et leitud väärtus oli lõppvastus, kuigi see seda ei ole, annab programm sellest teada vastava veateatega. Kui tegemist oli lõppvastusega, siis ülesande edukal lahendamisel näitab programm tehtud lahenduskäiku (Joonis 9).



Joonis 9. Lõpetatud ülesanne.

2.1.4. Lahendusfail

Ülesande lahendamise jooksul salvestatakse kasutaja tehtud käigud tekstifaili, mille nimi on kujul “<ülesande nimi>_<lahendaja nimi>.txt”, kus lahendaja nime osa võib olla tühi. Faili struktuur on järgmine:

- 1) ülesande algust tähistab “=UUS”, millele järgnevad lahendaja nimi, ülesande avamise kuupäev ja kellaaeg;
- 2) lahendusefaili nimi;
- 3) kasutaja tehtud lahenduskäik;
- 4) kui ülesanne on edukalt lahendatud, lisatakse rida, kus on kirjas “Suurepärane! Ülesanne lahendatud.”, millele järgnevad vastuse andmise kuupäev ning kellaaeg;
- 5) ülesande lahendamisel tehtud vigade arv, mis on jaotatud nelja gruppi: küsimuse koostamise, arvutamise, tehte koostamise ning vastuse andmise vead.

Kui sama nimega kasutaja lahendab üht ülesannet mitu korda, salvestatakse ta lahenduskäigud samasse faili eelmise lahenduse järele.

Näide lahendusfailist:

```
=UUS  Katrin 2014/05/09 13:58:14
mandariinid_Katrin
Mitu õpilast õpib teises klassis?
(a)+(b)
8.0 + 9.0 = 17.0
Mitu õpilast õpib kolmandas klassis?
(c)+(d)
7.0 + 10.0 = 17.0
Mitu poissi õpib kolmandas klassis?
Sellele küsimusele on juba vastus olemas.
Mitu õpilast õpib teises ja kolmandas klassis?
(c+d)+(a+b)
17.0 + 17.0 = 34.0
Mitu mandariini said õpilased teises ja kolmandas klassis?
(a+b+c+d)*(e)
34.0 * 2.0 = 68.0
Suurepärane! Ülesanne lahendatud. . 2014/05/09 13:59:19
Katrin: mandariinid_Katrin 1 0 0 0
```

2.2. Ülesande koostamine

Lahendamise programmi jaoks on võimalik koostada vajalikul kujul ülesannete tekstifaile kasutades tavalisi tekstiredaktoreid, kuid et vältida vigu ning teha protsess mugavamaks on Tekster-il ka ülesannete koostamise programm.

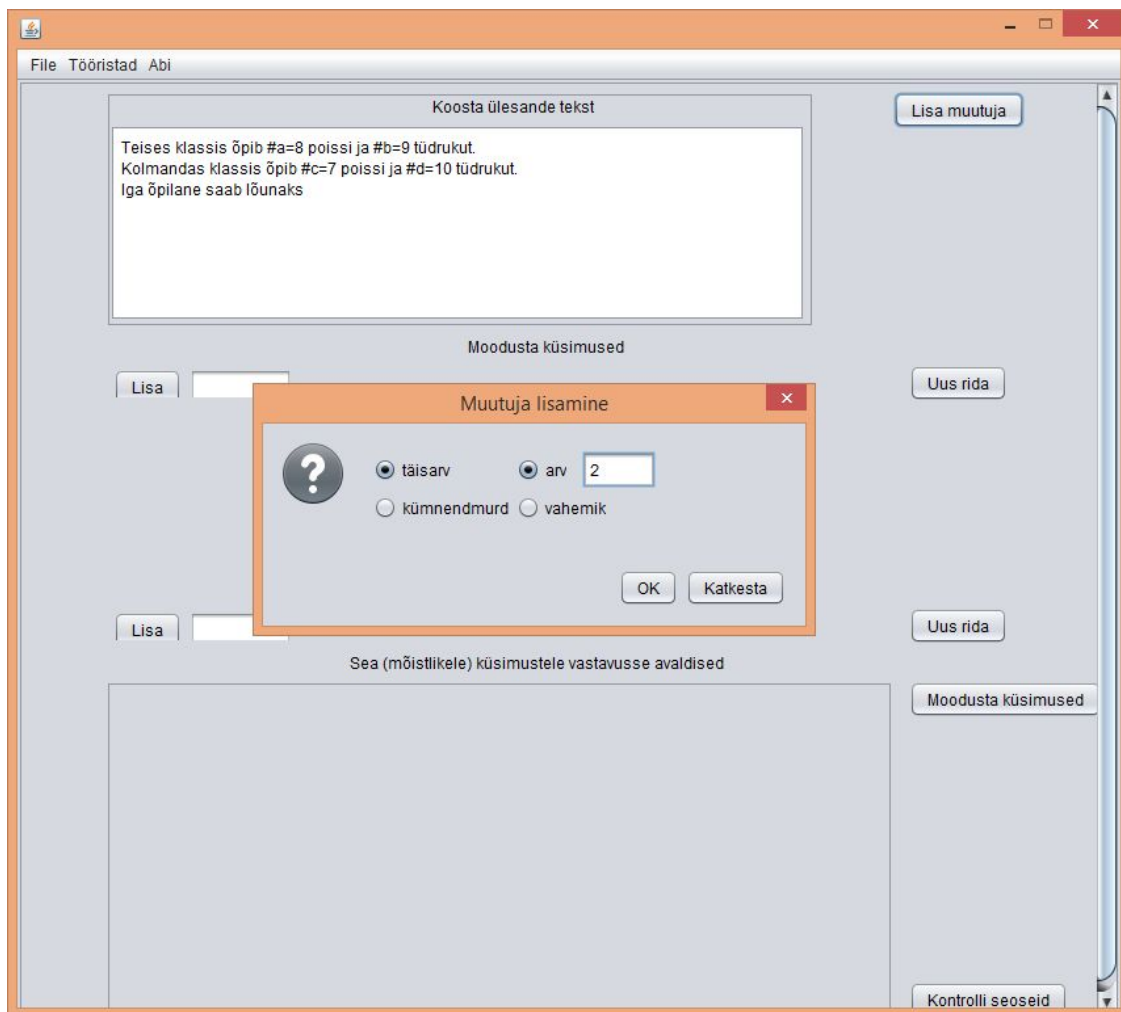
Ülesande koostamine koosneb kolmest etapist: ülesande teksti sisestamine, küsimuste moodustamine ning küsimustele vastavate avaldiste määramine.

2.2.1. Ülesande teksti sisestamine

Ülesandepüstitus tuleb kasutajal sisestada tekstikasti, mille pealkirjaks on “Koosta ülesande tekst”. Iga arvulise väärtuse sisestamiseks peab ülesande koostaja vajutama nupule “Lisa muutuja”. Esilekerkivas dialoogiaknas on vaja valida, mis tüüpi muutujaga on tegemist, ning sisestada vastav väärtus (Joonis 10). Dialoogiaknas olevast vasakust tulbast on vaja valida, mis tüüpi arve soovitakse sisestada (täisarv või kümnendmurd) ning paremast tulbast tuleb valida, kas valitud arvutüüpidest on vaja koostada vahemik või üksik arv. Kui sisestatakse arvude vahemik, lisatakse ülesande tekstifaili rida “!SEADED=”, millele järgneb rida, mille kuju on “#<muutuja tähis>:<vahemiku algus>:<vahemiku lõpp>:<1 või 0, vastavalt kas tegemist on kümnendmurdude või täisarvude vahemikuga>”, näiteks “#a:1:5:0”.

Muutujad lisatakse teksti kujul “#<tähis>=<väärtus>”, kus tähiseks on väike ladina täht. Kuna ladina tähestikus on 26 tähte, siis on ülesande tekstis esinev maksimaalne muutujate arv 26.

Kui kasutaja sisestab muutuja väärtuseks nulli, annab programm hoiatuse, kuna sellest tulenevalt võib tekkida ülesande lahendamisel nulliga jagamine.



Joonis 10. Ülesande teksti sisestamine.

2.2.2. Küsimuste moodustamine

Küsimuste moodustamiseks tuleb lause jagada osadeks ning need sisestada selleks ettenähtud lahtritesse (Joonis 11). Vasakul iga rea ees asetsevale nupule “Lisa” vajutades tekitatakse rea lõppu uus lahter, millesse on võimalik sisestada järgmine lauseosa. Paremal asuvale nupule “Uus rida” vajutades lisatakse viimase lauseosade rea alla uus rida. Veergudes asetsevatesse lahtritesse on võimalik sisestada lauseosi, millest lahendaja saab küsimuste koostamisel valida.

Küsimusi on võimalik moodustada kahes grupis. Mõlemas küsimustegrupis on võimalik moodustada kuni kuueelemendilisi lauseid ning iga lauseosa valikuks on võimalik määrata kuus erinevat võimalust.

Kui laused on moodustatud, peab kasutaja vajutama nupule “Moodusta küsimused”, mis genereerib kõik võimalikud küsimused järgmise etapi jaoks.

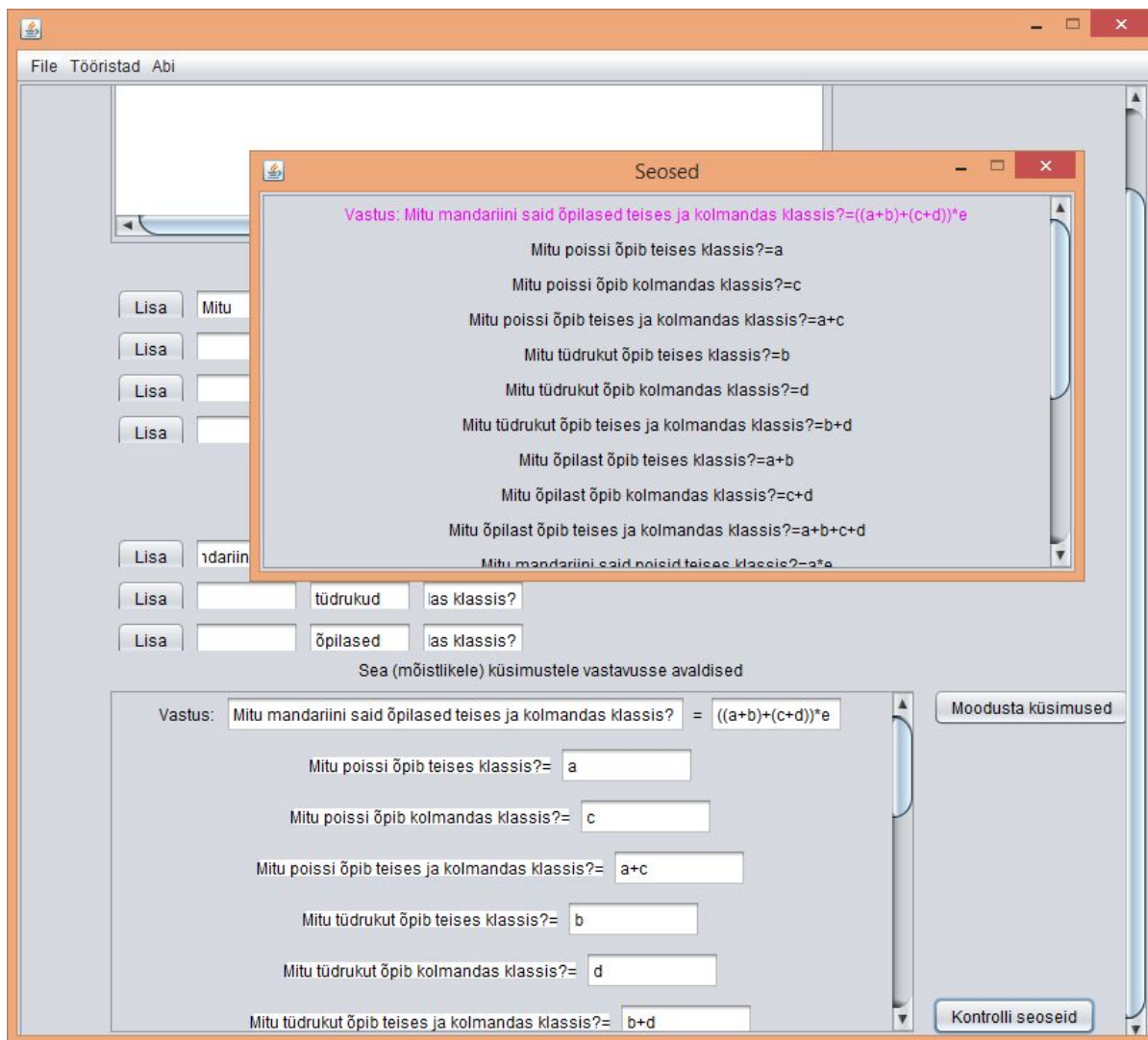
Moodusta küsimused				
Lisa	Mitu	poissi	õpib	teises klass
Lisa		tüdrukut		kolmandas
Lisa		õpilast		ses ja kolm:

Uus rida

Joonis 11. Küsimuste moodustamine.

2.2.3. Küsimustele vastavate avaldiste määramine

Nupule “Moodusta küsimused” vajutades genereerib programm kõikvõimalikud küsimused, mida lahendajal on võimalik moodustada. Ülesande koostaja peab igale mõistlikule küsimusele määrama sellele vastava avaldise, kasutades ülesande tekstis etteantud tähiseid. Mõttetute küsimuste avaldise lahter tuleb jätta tühjaks. Kasutaja peab ka ära määrama, millise küsimuse vastus on lõppvastus (Joonis 12).



Joonis 12. Koostatud ülesanne.

Kui küsimustele on avaldised määratud, tuleb koostajal vajutada nupule “Kontrolli seoseid”. Selle tulemusena kontrollib programm, kas lõppvastuse küsimus ja avaldis on määratud, kas avaldistes kasutatud muutujad eksisteerivad, kas sulge on võrdselt ning kas on kasutatud õigeid tehtemärke “+”, “-”, “*”, “/”. Kui ühtegi viga ei esine, kuvab programm uues aknas moodustatud küsimused ja neile vastavusse määratud avaldised. Pärast seda on võimalik ülesannet salvestada, mille tagajärjel koostatakse tekstifail, mida on võimalik avada ülesannete lahendamise programmiga.

Ülesande salvestamiseks, pooliku ülesande jätkamiseks ning uue ülesande alustamiseks on menüü “File” all vastavalt valikud “Salvesta”, “Ava” ning “Uus”.

2.2.4. Ülesande esitus tekstifailis

Koostatud ülesanne salvestatakse tekstifaili laiendiga “.txt”. Tekkinud faili struktuur on järgmine:

- 1) ülesanne algab reaga “-=TEKSTER2=-”;
- 2) muutujate kohta käiv informatsioon algab reaga “!SEADED=”. Muutujad on esitatud kujul <muutuja tähistus>:<vahemiku algus>:<vahemiku lõpp>:<1 või 0, vastavalt kas tegu on kümnendmurru või täisarvuga>. Kui ülesandes ei esine vahemikke, siis see osa failist puudub;
- 2) ülesande tekst algab reaga “!TEKST=”;
- 3) esimese küsimustegrupi maatriksi veergude osa algab reaga “!VEERUD=” ning teise grupi veergude osa reaga “!VEERUD2=”;
- 4) küsimuste ning nendele vastavusse määratud avaldiste osa algab reaga “!VASTAVUSED=”;
- 5) lõppvastusele vastav avaldis järgneb reale “!VASTUS=”.

Ülesande koostamise programm kontrollib vaid, kas esimese rea sisuks on “-=TEKSTER2=-”.

Näide ülesande esitusest tekstifailis:

```
-=TEKSTER2=-
!TEKST=
Teises klassis õpib #a=8 poissi ja #b=9 tüdrukut. Kolmandas klassis õpib #c=7 poissi ja #d=10
tüdrukut. Iga õpilane saab lõunaks #e=2 mandariini. Mitu mandariini said õpilased kokku?
!VEERUD=
Mitu
poissi&tüdrukut&õpilast
õpib
teises klassis?&kolmandas klassis?&teises ja kolmandas klassis?
!VEERUD2=
Mitu mandariini said
poisid&tüdrukud&õpilased
teises klassis?&kolmandas klassis?&teises ja kolmandas klassis?
!VASTAVUSED=
a=Mitu poissi õpib teises klassis?
c=Mitu poissi õpib kolmandas klassis?
a+c=Mitu poissi õpib teises ja kolmandas klassis?
```

b =Mitu tüdrukut õpib teises klassis?
 d =Mitu tüdrukut õpib kolmandas klassis?
 $b+d$ =Mitu tüdrukut õpib teises ja kolmandas klassis?
 $a+b$ =Mitu õpilast õpib teises klassis?
 $c+d$ =Mitu õpilast õpib kolmandas klassis?
 $a+b+c+d$ =Mitu õpilast õpib teises ja kolmandas klassis?
 $a*e$ =Mitu mandariini said poisid teises klassis?
 $c*e$ =Mitu mandariini said poisid kolmandas klassis?
 $(a+c)*e$ =Mitu mandariini said poisid teises ja kolmandas klassis?
 $b*e$ =Mitu mandariini said tüdrukud teises klassis?
 $d*e$ =Mitu mandariini said tüdrukud kolmandas klassis?
 $(b+d)*e$ =Mitu mandariini said tüdrukud teises ja kolmandas klassis?
 $(a+b)*e$ =Mitu mandariini said õpilased teises klassis?
 $(c+d)*e$ =Mitu mandariini said õpilased kolmandas klassis?
 $((a+b)+(c+d))*e$ =Mitu mandariini said õpilased teises ja kolmandas klassis?
 !VASTUS=
 $((a+b)+(c+d))*e$

3. Bakalaureusetöö käigus tehtud muudatused

Käesoleva bakalaureusetöö käigus lisati rakendusele järgnevad võimalused:

- Maxima - programmil on võimalik suhelda arvutialgebrasüsteemiga Maxima, mida rakendatakse ülesande automaatlahendamisel, kasutaja poolt koostatud avaldise võrduse kontrollil käesolevale küsimusele ülesande tekstifailis vastavusse seatud avaldisega ning kontrollil, kas kasutaja poolt moodustatud avaldis on ülesande lõppvastus.
- Automaatlahendus - programm on võimeline ülesannet algusest lõpuni lahendama, mida rakendatakse ülesannete lahendamise programmis vihje andmisel ning ülesannete koostamise programmis lahenduvuse kontrollil.
- Üleliigsete sammude kustutamine - võimalus ülesande lahendamise käigus tehtud üleliigseid samme kustutada, kui ülesanne on lõpuni lahendatud.

3.1. Maxima

Ülesannete koostamise ja lahendamise programmide tööks peab kasutajal olema installeeritud arvutialgebrasüsteemi Maxima [5] versioon 5.16.x või uuem.

Maxima ühendamiseks Tekster-iga on kasutatud avatud lähtekoodiga projekti Jacomax. Jacomax-i koodist on eemaldatud meetodid, mis on seotud logimisega. Ka on muudetud Maxima otsimist kasutaja arvutist nii, et kui nimekirjas ettetulev kaust kasutajal eksisteerib, siis kontrollitakse, kas vastavas kaustas on Maxima programm või mitte ning kui ei ole, siis liigutakse järgmise juurde. Seega peaks Tekster-i käivitamiseks olema kasutaja arvutisse paigaldatud Maxima soovitatavalt kausta, mida installeerimisel vaikumisi pakutakse. Näiteks Windows operatsioonisüsteemide puhul on nendeks kaustadeks “Program Files” või “Program Files (x86)”.

Maxima käskudest on Tekster-is kasutatud järgmisi:

- 1) “is(equal(*avaldis*₁, *avaldis*₂))”, mis kontrollib, kas kaks sisestatud avaldist on võrdsed;
- 2) “expand(*avaldis*)”, mis avab sulud ning viib avaldised kindlale kujule.

Peale ülesande automaatse lahendamise, kasutatakse Maximat ka kontrollimaks, kas kasutaja koostatud avaldis on võrdne moodustatud küsimusele määratud avaldisega ülesande tekstifailis, ning kas lahendaja on leidnud lõppvastuse või mitte.

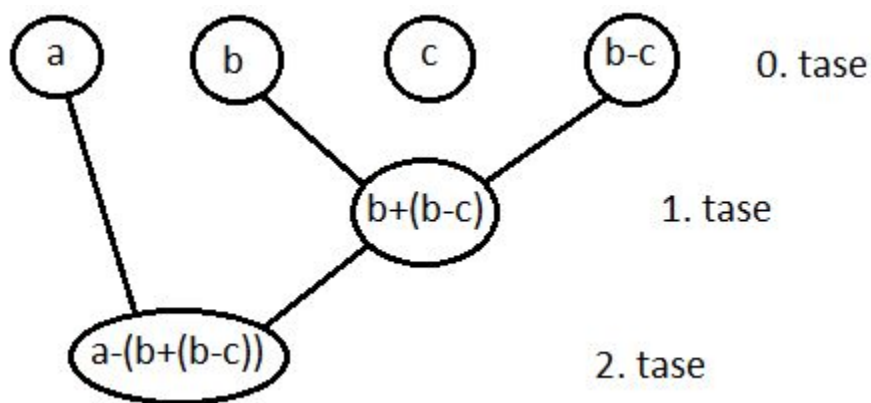
3.2. Ülesande automaatne lahendamine

Lahendamise põhimõte jälgib graafi moodustamist ning selle tippudest lõppvastuse otsimist. Graaf koosneb tippudest, mida võivad omavahel ühendada servad.

Otsingu realiseerimiseks on kasutatud iga graafi tipu jaoks struktuuri, mis sisaldab järgmist infot:

- 1) Tipus esinev avaldis - avaldis, mis viimasena moodustati, ning selle kuju pärast Maxima käsu “expand()” rakendamist;
- 2) Käesolevas tipus teadaolevad väärtused - avaldised, mida on kasutatud, et vastavasse tippu jõuda, ning nende kuju pärast Maxima käsu “expand()” rakendamist.

Järgnevalt on nimetatud nulltasemeks esimesel sammul leitud tippe, ning igaks järgmiseks tasemeks eelmise taseme tippude naabreid ning tippe, mis on moodustatud kasutades eelmise taseme tippe koos mõne eelmisest tasemest eespool asuva taseme tipuga (Joonis 13). Järgnevalt kirjeldatud algoritmi kohaselt otsitakse lõppvastust sisaldavat tippu ühelt tasemelt ning kui vastaval tasemel lõppvastust ei esine, moodustatakse järgmine tase, millest lõppvastust otsida. Algoritmi töö lõppeb, kui otsitud tipp on leitud või kogu graaf on läbitud. Käesoleva otsingualgoritmiga leitakse ülesandele lühim võimalik lahendus.



Joonis 13. Moodustatud graaf lõppvastuse “a-(b+b-c)” puhul.

Lahenduse käigus salvestatakse kõik senimaani leitud avaldised ühte kogumikku. Graafi iga järgmise taseme tippude konstrueerimiseks võetakse vastavaks hetkeks leitud avaldised, lõppvastuses esinevad tehtemärgid (kuna kõiki nelja tehet liitmine, jagamine, korrutamine, lahutamine ei pruugi olla kasutatud, seega neid ei ole mõttekas uurida) ja moodustatakse kõikvõimalikud kombinatsioonid kujul <avaldis><tehtemärk><avaldis>. Kombineerimisel peetakse silmas seda, et liitmise ja korrutamise puhul ei ole tehte järjekord oluline, kuid jagamise ning lahutamise puhul on, ning seda, et ei moodustataks kombinatsioone, mida eelmiste tasemete leidmiseks juba on uuritud. Saadud kombinatsioonist moodustatakse uus tipp vaid siis, kui moodustatud avaldisele on ülesandefailis määratud vastav küsimus ehk koostatud avaldis on sisukas.

Ülesande lahenduse algoritm on järgmine:

1. Moodustatakse nimekiri lahenduse käigus leitud tippude avaldistest. Kui ülesannet lahendatakse algusest, on selles nimekirjas vaid lõppvastuses esinevad muutujad. Kui ülesannet lahendatakse mingist kasutaja seisust, on selles nimekirjas lõppvastuses esinevad muutujad ning kasutaja poolt leitud avaldised.
2. Moodustatakse lahenduse käigus leitud tippude avaldistest, mis asuvad 1. punktis moodustatud nimekirjas, kombinatsioonid eelpool kirjeldatud viisil. Sisukatest avaldistest moodustatakse tipud. Kui koostatakse nulltasandit, moodustatakse tipud ka kasutaja leitud väärtustele vastavatest avaldistest ning lõppvastuses esinevatest muutujatest. Kõik konstrueeritud tipud kokku moodustavad uue tasandi ning tippude avaldised lisatakse leitud avaldiste nimekirja. Kui koostatud tasandil leidub lõppvastus või tasand on tühi (ehk ükski moodustatud avaldis ei olnud sisukas), liigutakse viimase sammu juurde. Vastasel juhul korratakse käesolevat sammu.

Kui tipuks on lõppvastuses esinev muutuja, on tipu andmed järgmised:

- 1) tipus esinev avaldis - muutuja;
- 2) tipus teadaolevad väärtused - tühi.

Kui tipuks on kasutaja poolt leitud avaldis, on tipu andmed järgmised:

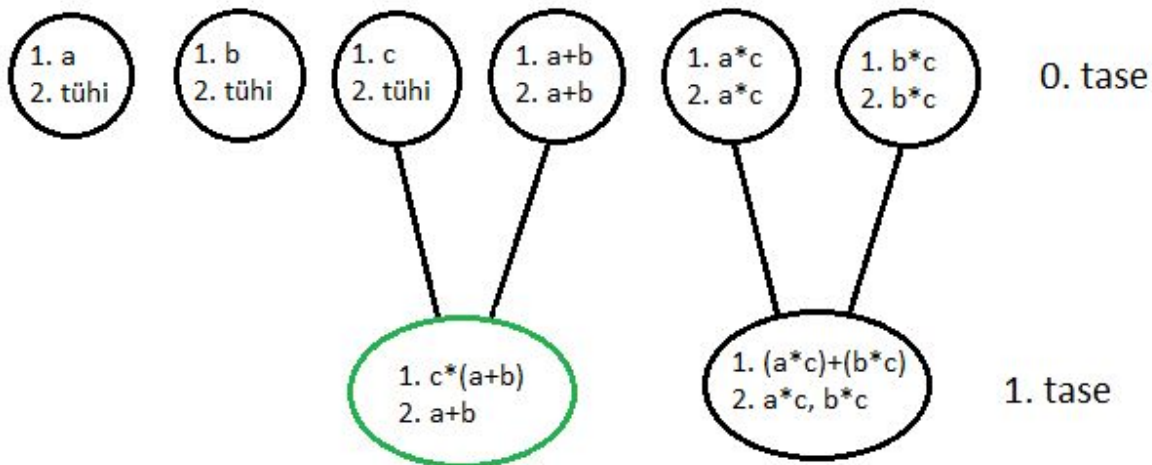
- 1) tipus esinev avaldis - kasutaja poolt tehtud avaldis;

2) tipus teadaolevad väärtused - tühi.

Kõikide ülejäänud tippude andmed on järgmised:

- 1) tipus esinev avaldis - leitud avaldis;
 - 2) tipus teadaolevad muutujad - leitud avaldis ja tippudes, millest käesolev tipp moodustati, teadaolevad avaldised kordusteta.
3. Koostatakse kogum viimasena leitud tasandil esinevatest tippudest, milles leidub lõppvastus. Kogumist valitakse tipp, mille teadaolevate avaldiste nimekiri on lühim. Lahenduskäik koosneb vastavas nimekirjas leitud avaldistest. Kui lõppvastust sisaldavate tippude kogum on tühi, ei ole ülesanne lahenduv.

Järgnevalt on näha algoritmi moodustatud puud, kui otsitakse lahendust ülesandele, mille lõppvastus on $(a+b)*c$ (Joonis 14) ning mille ülesandefailis on sisukateks määratud järgnevad avaldised: $a+b$, $a*c$, $b*c$ ja $(a+b)*c$. Roheline tipp tähistab lõpptippu, mille lahendustee on lühim ning mis valitakse ülesande lahenduseks.



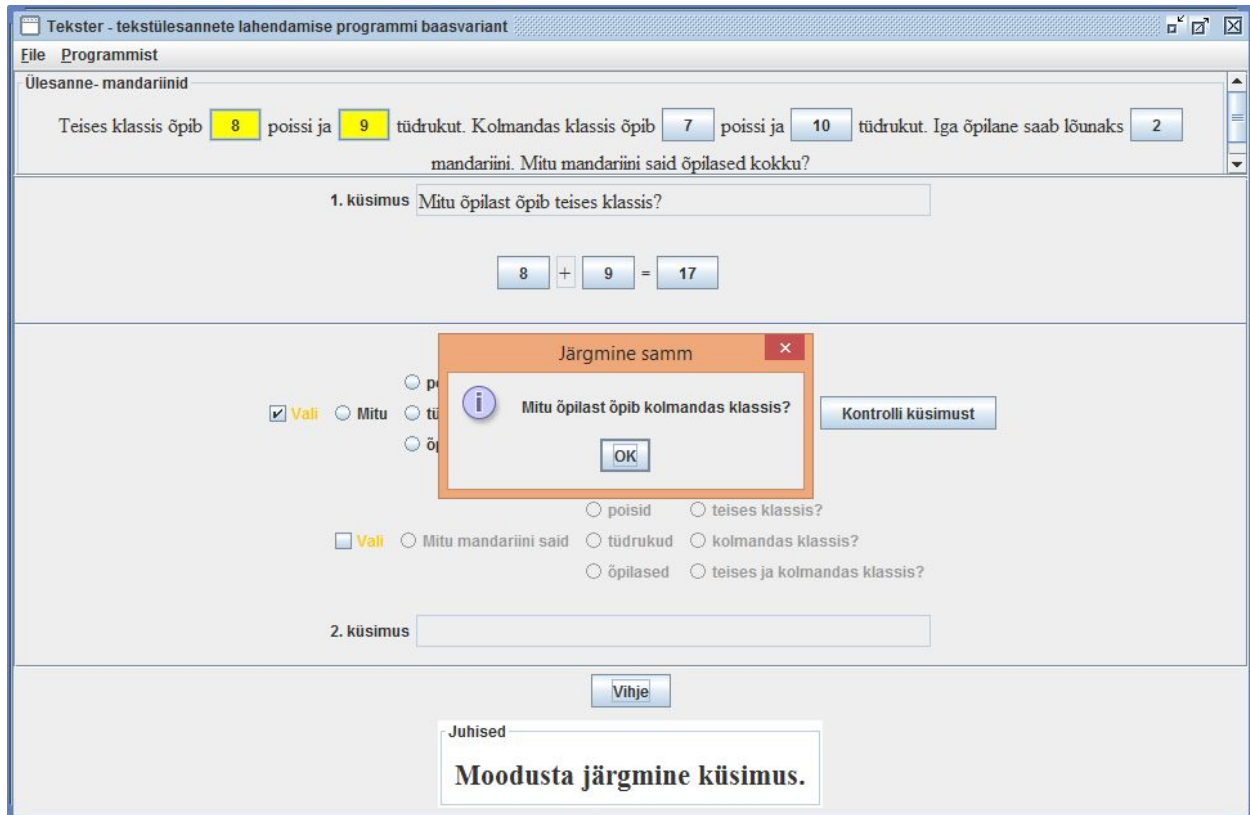
Joonis 14. Lahenduskäik lõppvastuse $(a+b)*c$ puhul.

3.2.1. Automaatlahenduse rakendamine ülesannete lahendamise programmis

Tekster-i ülesannete lahendamise programmi on lisatud võimalus lahenduse käigus küsida programmilt, millist küsimust järgmisena koostada. Vihje andmine toimub järgmiselt:

- 1) Kasutaja vajutab nupule "Vihje".

- 2) Programm lahendab ülesande lõpuni eelpool kirjeldatud algoritmi järgi.
- 3) Programm koostab nimekirja avaldistest, mille sooritamine on vajalik ülesande lõpule viimiseks, kuid mida kasutaja pole veel teinud. Sellest nimekirjast valitakse esimene avaldis ehk tipp, mis asub väikseima numbriga tasandil.
- 4) Kasutajale väljastatakse küsimus, mis vastab eelmisel sammul leitud avaldisele (Joonis 15).



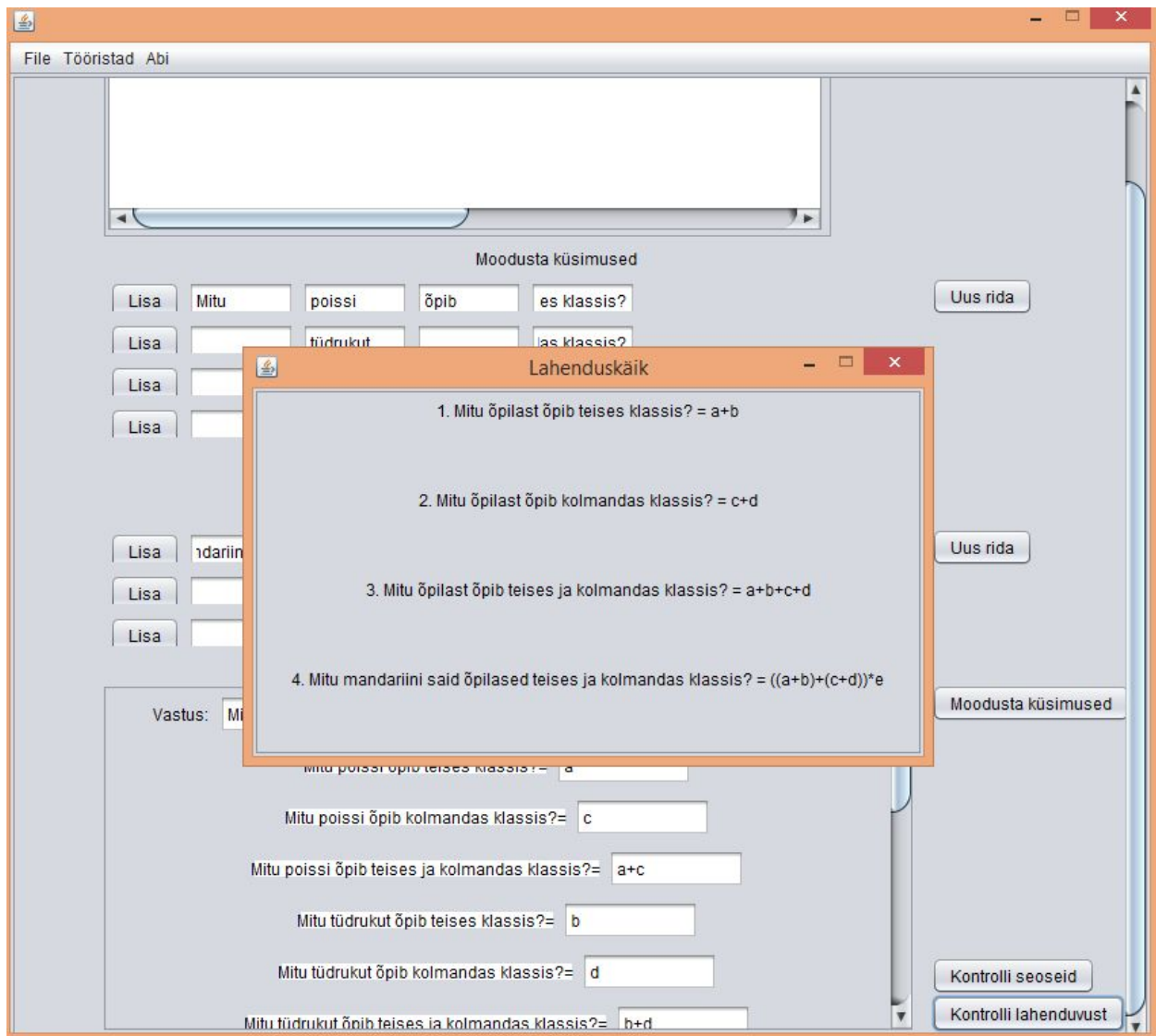
Joonis 15. Vihje andmine.

3.2.2. Automaatlahenduse rakendamine ülesannete koostamise programmis

Tekster-i ülesannete koostamise programmi on lisatud võimalus kontrollida, kas koostatud ülesanne on lahenduv. Lahenduvuse kontrollimine toimub järgmiselt:

- 1) Kasutaja on koostanud ülesande ning vajutanud nupule “Kontrolli seoseid”. Selle tagajärjel saab programm sisendiks küsimuste ja neile vastavate avaldiste paarid, mis kasutaja on seadnud ning mille abil on võimalik ülesannet lahendada.

- 2) Kui seoste kontroll on edukas, on kasutajal võimalik vajutada nupule “Kontrolli lahenduvust”. Selle tulemusena lahendab programm sisestatud ülesande eespool kirjeldatud algoritmi järgi.
- 3) Kui ülesanne on lahenduv ehk algoritm leidis ülesandele vastava lahenduskäigu, väljastab programm kasutajale ülesande lahenduskäigu (Joonis 16). Vastasel juhul, kui programm lõpetab töö ilma lahenduskäiku leidmata, antakse veateade.



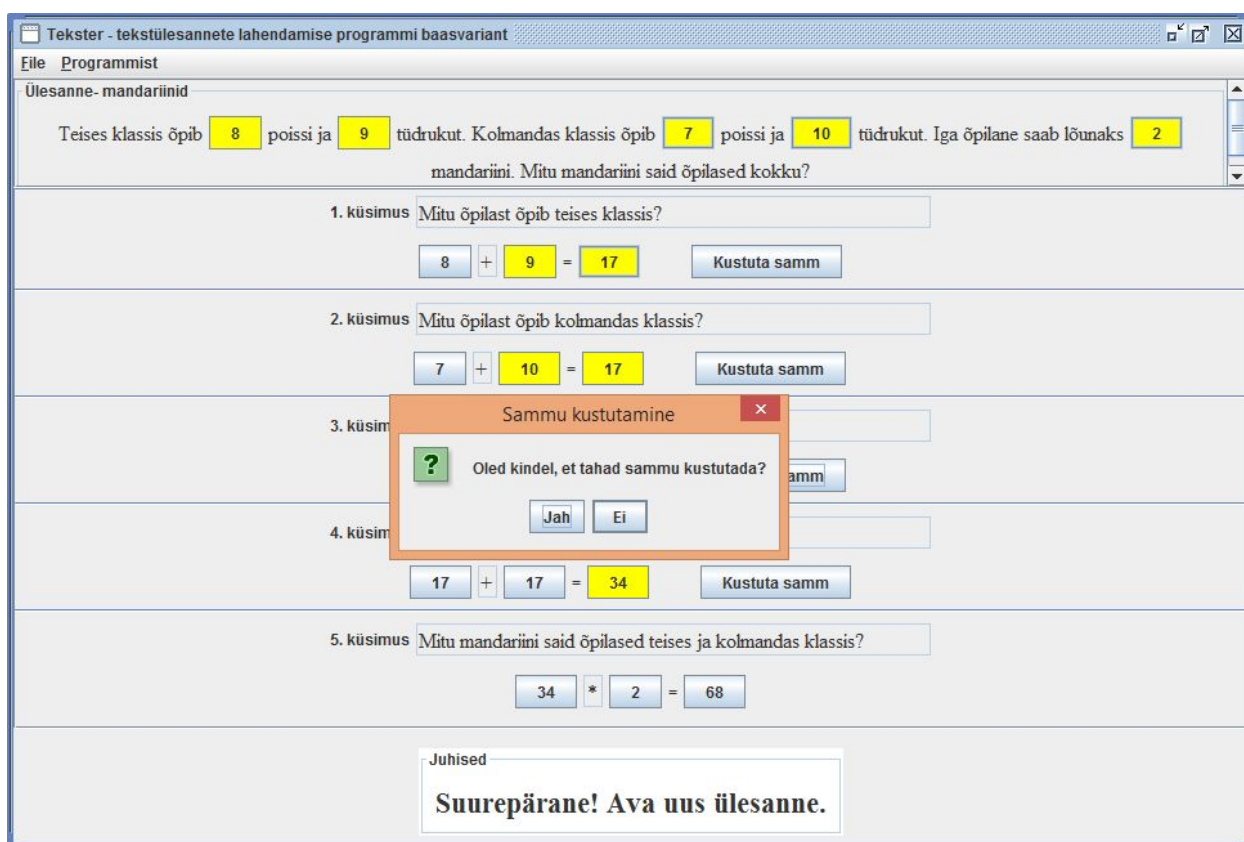
Joonis 16. Lahenduva ülesande lahenduskäik.

Ülesande mittelahenduvuse peamisteks põhjusteks on:

- 1) Lõpplahenduseni jõudmiseks pole koostatud piisavalt vajalikke vahevastuseid;
- 2) Avaldised on vigaselt moodustatud.

3.3. Sammude kustutamine

Kui ülesanne on edukalt lõpuni lahendatud, on kasutajal võimalus kustutada oma lahendusteest samme, mida lõppvastuseni jõudmiseks vaja ei olnud. Selleks on lahendatud ülesande puhul lisatud iga väärtuse, väljaarvatud lõppvastuse, järele nupp “Kustuta samm”. Nupule vajutades peab kasutaja valima, kas tõesti soovib sammu kustutada (Joonis 17). Kui lahendaja tahab käiku kustutada, kuid seda on kasutatud lõpplahenduse saamiseks, teatab programm, et valitud sammu ei ole võimalik kustutada. Vastasel juhul kustutatakse samm.



Joonis 17. Sammu kustutamine.

Kui sammu kustutamine õnnestub, lisatakse lahenduse faili rida “Kustutati samm: <avaldis>”, näiteks “Kustutati samm: a+b”.

4. Programmi edasiarendamise võimalused

Lahendamise programmile on võimalus lisada:

- 1) Pikemate avaldiste moodustamine - käesolevas versioonis on võimalik moodustada vaid tehteid;
- 2) Konstantide kasutamine - leidub ülesandeid, mille lahendamine nõuab väärtuste kasutamist, mida tingimata ülesande tekstis ei esitata, näiteks π ;
- 3) Õpilase lahenduskäigu hindamine - võrreldakse lühimat ja kasutaja moodustatud lahendust;
- 4) Negatiivsete arvudega arvutamine;
- 5) Lahenduskäigu faili krüptimine - käesolevas versioonis saab õpilane lihtsalt faili muuta;
- 6) Küsimustegrupi lisamine - hetkel on gruppide maksimaalne arv kaks.
- 7) Illustreeriva informatsiooni lisamine - leidub tekstülesandeid, mille osa informatsioonist on edasi antud näiteks jooniste kaudu.

Koostamise programmile on võimalus lisada:

- 1) Negatiivsete arvude kasutamine;
- 2) Lahenduskäigu vigade analüüsimine - tehtud vigade statistika koostamine;
- 3) Lahenduskäigu ja tehtud vigade kuvamine - käesolevas versioonis on vead ja lahenduskäik nähtavad tekstifailis, koostamise programmiga võiks olla neid võimalik mugavamalt ning arusaadavamalt näha.
- 4) Lahenduskäigu kontrollimine - võimalus tuvastada pärast lahenduse lõpetamist tehtud muudatusi lahenduskäigu failis.

Kokkuvõte

Käesoleva töö eesmärgiks oli 2006. aastal loodud ning 2013. aastal täiendatud tekstülesannete programmi Tekster täiendamine. Töö käigus lisati programmile järgmised võimalused:

- Maxima - programmil on võimalik suhelda arvutialgebrasüsteemiga Maxima, mida rakendatakse ülesande automaatlahendamisel.
- Ülesande automaatlahendus - programm oskab etteantud ülesannet algusest lõpuni lahendada ning lahenduskäigu tagastada.
- Ülesande lahendamise käigus vihje, millist küsimust järgmisena koostada, pakkumine.
- Ülesande koostamisel moodustatud ülesande lahenduvuse kontroll.
- Üleliigsete sammude kustutamine - võimalus ülesande lahendamise käigus tehtud üleliigseid samme kustutada, kui ülesanne on lõpuni lahendatud.

Samuti sai töö käigus uuritud Tekster-iga sarnaseid programme, mida võib silmas pidada Tekster-i edasisel arendamisel.

Tsiteeritud teosed

Kõik viited on kontrollitud 14.05.2014 seisuga.

- [1] 1. E. Koppel. Tekstülesannete lahendamise programmi baasvariant. Bakalaureusetöö. 2006.
- [2] J. Kibal. Tekstülesannete lahendamise programmi funktsionaalsuse täiendamine . Bakalaureusetöö. 2013.
- [3] *CARNEGIE LEARNING MATHia SOFTWARE*;
<http://mathseries.carnegielearning.com/product-info/software/>
- [4] 2. *Thinking Blocks - Model and Solve Math Word Problems*;
<http://www.mathplayground.com/thinkingblocks.html>
- [5] *Maxima, a Computer Algebra System*;
<http://maxima.sourceforge.net/>

Lisad

I. Projekt

1. Programmi kood;
2. Ülesannete kogu.

II. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Katrin Valdson

(sünnikuupäev: 07.08.1992)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose, “Tekstülesannete lahendamise programmi täiendamine”, mille juhendaja on Rein Prank,
 - 1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 16.05.2014.